

CLAIMS

1. A secure processor for a computing device, the processor being operable in a normal mode and a preferred mode, the processor including a security kernel for being instantiated on the processor when the processor enters into the preferred mode and a security key accessible by the instantiated security kernel when the processor is operating in the preferred mode, the security kernel employing the accessed security key during the preferred mode to authenticate a secure application on the computing device, wherein the security kernel allows the processor to be trusted to keep hidden a secret of the application.
2. The processor of claim 1 in combination with the application.
3. The processor of claim 2 wherein the application is selected from a group consisting of a digital rights management (DRM) system and a banking / financial system.
4. The processor of claim 1 wherein the security kernel automatically authenticates a particular application.
5. The processor of claim 1 wherein the security kernel initially authenticates a chooser application that allows a user to select from at least one available applications on the computing device.
6. The processor of claim 1 wherein the security kernel employs the accessed security key during the preferred mode to decrypt at least one encrypted key for the application.

7. The processor of claim 1 in combination with the computing device.

8. The processor of claim 7 wherein the computing device is a portable computing device.

9. The processor of claim 1 further comprising a storage space, the security kernel being permanently stored in the storage space.

10. The processor of claim 1 further comprising a storage space, the security key being permanently stored in the storage space.

11. The processor of claim 1 wherein the security kernel employs the accessed security key during the preferred mode to authenticate / verify the application prior to instantiation thereof.

12. The processor of claim 11 wherein the security kernel performs a hash / MAC (message authentication code) over at least a portion of the application and then compares the hash / MAC to a hash / MAC corresponding to the application.

13. The processor of claim 1 wherein such processor enters preferred mode whenever a predefined initializing processor action is performed.

14. The processor of claim 13 wherein such processor enters preferred mode whenever a CPU reset is performed.

15. A method for a secure processor to instantiate and authenticate a secure application thereon by way of a security kernel, the method comprising:

entering a preferred mode where a security key of the processor is accessible;

instantiating and running a security kernel, the security kernel:

accessing the security key;

applying the accessed security key to decrypt at least one encrypted key for the application;

storing the decrypted key(s) in a location where the application will expect the key(s) to be found; and

authenticating the application on the processor; and

entering a normal mode from the preferred mode after the security kernel authenticates the application, where the security key is not accessible;

wherein the security kernel allows the processor to be trusted to keep hidden the key(s) of the application.

16. The method of claim 15 wherein entering the preferred mode comprises entering the preferred mode upon a CPU reset.

17. The method of claim 15 further comprising erasing data in a cache of the processor prior to instantiating the security kernel.

18. The method of claim 15 further comprising erasing data in a cache of the processor after entering normal mode.

19. The method of claim 15 wherein the security kernel employs the accessed security key during the preferred mode to authenticate / verify the application prior to instantiation thereof.

20. The method of claim 19 wherein the security kernel performs a hash / MAC (message authentication code) over at least a portion of the

application and then compares the hash / MAC to a hash / MAC corresponding to the application.

21. The method of claim 15 wherein the security key of the processor is a symmetric key and the application is instantiated from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is the security key, KMAN is a device key of the portable device independent of the security key, and KCODE is the secret of the application, and wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

- applying KCPU to KCPU (KMAN) to produce KMAN; and
- applying KMAN to KMAN (KCODE) to produce KCODE.

22. The method of claim 21 wherein the security key of the processor is a symmetric key and the application is instantiated from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
MAC (main body, KMAN)	message authentication code of the main body under KMAN
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is the security key, KMAN is a device key of the portable device independent of the security key, and KCODE is the secret of the application, and wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

- applying KCPU to KCPU (KMAN) to produce KMAN;

computing MAC (main body, KMAN);
comparing the computed MAC to MAC (main body, KMAN)
from the header to determine if the code image has been changed; and
if the MACs match, applying KMAN to KMAN (KCODE) to
produce KCODE.

23. The method of claim 15 wherein the security key of the
processor is a private key of a public key - private key pair and the application is
instantiated from a code image including a main body and a header including:

public key (KCODE)	KCODE encrypted according to the public key
--------------------	--

where KCODE is the secret of the application, and
wherein the security kernel applying the accessed security key to
decrypt at least one encrypted key for the application comprises applying the
security key as the private key to public key (KCODE) to produce KCODE.

24. The method of claim 23 wherein the security key of the
processor is a private key of a public key - private key pair and the application is
instantiated from a code image including a main body and a header including:

public key (HASH (main body), KCODE)	Hash of the main body and KCODE, both encrypted according to the public key
---	---

where KCODE is the secret of the application, and
wherein the security kernel applying the accessed security key to
decrypt at least one encrypted key for the application comprises:
computing HASH (main body);

applying the private key to public key (HASH (main body), KCODE) to produce HASH (main body) and KCODE;

comparing the computed HASH to the produced HASH to determine if the code image has been changed;; and

if the HASHs match, employing the produced KCODE as appropriate.

25. A method for a secure processor to instantiate one of a plurality of available secure applications thereon by way of a security kernel, the method comprising:

setting a chooser value to a value corresponding to a chooser application upon power-up;

entering a preferred mode upon a power-up CPU reset and instantiating the security kernel, the security kernel determining that the chooser value corresponds to the chooser application and therefore authenticating same, the chooser application being instantiated;

entering a normal mode after the chooser application is instantiated and leaving same to run, the chooser application presenting the plurality of available applications for selection by a user;

receiving a selection of one of the presented applications to be instantiated;

setting the chooser value to a value corresponding to the selected application;

entering a preferred mode upon an executed CPU reset and instantiating the security kernel, the security kernel determining that the chooser value corresponds to the selected application and therefore authenticating same, the selected application being instantiated;

entering a normal mode after the selected application is instantiated and leaving same to run;

wherein the security kernel allows the processor to be trusted to keep hidden a secret of the chooser application and a secret of the selected application.

26. The method of claim 25 further comprising setting the chooser value to the value corresponding to the chooser application upon the selected application being authenticated by the security kernel, wherein upon execution of a CPU reset, the security kernel determines that the chooser value corresponds to the chooser application 72c and therefore authenticates same.

27. The method of claim 25 further comprising storing the chooser value in a memory location not affected by a CPU reset so that the stored chooser value is available after same.

28. A method for a secure processor to instantiate a secure application thereon, the method comprising:

instantiating a first security kernel which employs symmetric cryptography;

instantiating by way of the instantiated first security kernel a second security kernel which employs asymmetric cryptography; and

authenticating by way of the instantiated second security kernel the secure application.

29. The method of claim 28 wherein the security key of the processor is a symmetric key and the second security kernel is instantiated by the first security kernel from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is a security key of the processor, KMAN is a device key independent of the security key, and KCODE is the private key of the second security kernel, and

wherein the first security kernel applies the security key to decrypt the private key of the second security kernel during instantiation thereof by:

applying KCPU to KCPU (KMAN) to produce KMAN; and
applying KMAN to KMAN (KCODE) to produce KCODE.

30. The method of claim 29 wherein the application is instantiated by the second security kernel from a code image including a main body and a header including:

public key (KCODE)	KCODE encrypted according to the public key
--------------------	---

where KCODE is the secret of the application, and

wherein the second security kernel applies the private key to decrypt the secret of the application during authentication thereof.

31. A computer-readable medium having stored thereon computer-executable instructions implementing a method for a secure processor to instantiate a secure application thereon by way of a security kernel, the method comprising:

entering a preferred mode where a security key of the processor is accessible;

instantiating and running a security kernel, the security kernel:

accessing the security key;

applying the accessed security key to decrypt at least one encrypted key for the application;

storing the decrypted key(s) in a location where the application will expect the key(s) to be found; and

authenticating the application on the processor; and
entering a normal mode from the preferred mode after the
security kernel authenticates the application, where the security key is not
accessible;

wherein the security kernel allows the processor to be trusted to
keep hidden the key(s) of the application.

32. The medium of claim 31 wherein entering the preferred mode
comprises entering the preferred mode upon a CPU reset.

33. The medium of claim 31 wherein the method further
comprises erasing data in a cache of the processor prior to instantiating the
security kernel.

34. The medium of claim 31 wherein the method further
comprises erasing data in a cache of the processor after entering normal mode.

35. The medium of claim 31 wherein the security kernel employs
the accessed security key during the preferred mode to authenticate / verify the
application prior to instantiation thereof.

36. The medium of claim 35 wherein the security kernel performs
a hash / MAC (message authentication code) over at least a portion of the
application and then compares the hash / MAC to a hash / MAC corresponding to
the application.

37. The medium of claim 31 wherein the security key of the
processor is a symmetric key and the application is instantiated from a code
image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is the security key, KMAN is a device key of the portable device independent of the security key, and KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

applying KCPU to KCPU (KMAN) to produce KMAN; and
applying KMAN to KMAN (KCODE) to produce KCODE.

38. The medium of claim 37 wherein the security key of the processor is a symmetric key and the application is instantiated from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
MAC (main body, KMAN)	message authentication code of the main body under KMAN
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is the security key, KMAN is a device key of the portable device independent of the security key, and KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

applying KCPU to KCPU (KMAN) to produce KMAN;
computing MAC (main body, KMAN);
comparing the computed MAC to MAC (main body, KMAN)
from the header to determine if the code image has been changed; and
if the MACs match, applying KMAN to KMAN (KCODE) to produce KCODE.

39. The medium of claim 31 wherein the security key of the processor is a private key of a public key - private key pair and the application is instantiated from a code image including a main body and a header including:

public key (KCODE)	KCODE encrypted according to the public key
--------------------	---

where KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises applying the security key as the private key to public key (KCODE) to produce KCODE.

40. The medium of claim 39 wherein the security key of the processor is a private key of a public key - private key pair and the application is instantiated from a code image including a main body and a header including:

public key (HASH (main body), KCODE)	Hash of the main body and KCODE, both encrypted according to the public key
--------------------------------------	---

where KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

- computing HASH (main body);
- applying the private key to public key (HASH (main body), KCODE) to produce HASH (main body) and KCODE;
- comparing the computed HASH to the produced HASH to determine if the code image has been changed;; and
- if the HASHs match, employing the produced KCODE as appropriate.

41. A computer-readable medium having computer-executable instructions thereon implementing a method for a secure processor to instantiate one of a plurality of available secure applications thereon by way of a security kernel, the method comprising:

setting a chooser value to a value corresponding to a chooser application upon power-up;

entering a preferred mode upon a power-up CPU reset and instantiating the security kernel, the security kernel determining that the chooser value corresponds to the chooser application and therefore authenticating same, the chooser application being instantiated;

entering a normal mode after the chooser application is instantiated and leaving same to run, the chooser application presenting the plurality of available applications for selection by a user;

receiving a selection of one of the presented applications to be instantiated;

setting the chooser value to a value corresponding to the selected application;

entering a preferred mode upon an executed CPU reset and instantiating the security kernel, the security kernel determining that the chooser value corresponds to the selected application and therefore authenticating same, the selected application being instantiated;

entering a normal mode after the selected application is instantiated and leaving same to run;

wherein the security kernel allows the processor to be trusted to keep hidden a secret of the chooser application and a secret of the selected application.

42. The medium of claim 41 wherein the method further comprises setting the chooser value to the value corresponding to the chooser application upon the selected application being authenticated by the security kernel, wherein upon execution of a CPU reset, the security kernel determines

that the chooser value corresponds to the chooser application 72c and therefore authenticates same.

43. The medium of claim 41 wherein the method further comprises storing the chooser value in a memory location not affected by a CPU reset so that the stored chooser value is available after same.

44. A computer-readable medium having stored thereon computer-executable instructions implementing a method for a secure processor to instantiate a secure application thereon, the method comprising:
instantiating a first security kernel which employs symmetric cryptography;
instantiating by way of the instantiated first security kernel a second security kernel which employs asymmetric cryptography; and
authenticating by way of the instantiated second security kernel the secure application.

45. The medium of claim 44 wherein the security key of the processor is a symmetric key and the second security kernel is instantiated by the first security kernel from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is a security key of the processor, KMAN is a device key independent of the security key, and KCODE is the private key of the second security kernel, and
wherein the first security kernel applies the security key to decrypt the private key of the second security kernel during instantiation thereof by:
applying KCPU to KCPU (KMAN) to produce KMAN; and

applying KMAN to KMAN (KCODE) to produce KCODE.

46. The medium of claim 45 wherein the application is instantiated by the second security kernel from a code image including a main body and a header including:

public key (KCODE)	KCODE encrypted according to the public key
--------------------	---

where KCODE is the secret of the application, and

wherein the second security kernel applies the private key to decrypt the secret of the application during instantiation thereof.

	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---